

**REMARKS**

The foregoing Amendment and the following Remarks are submitted in response to the Office Action issued on January 18, 2006 in connection with the above-identified patent application, and are being filed within the three-month shortened statutory period set for a response by the Office Action.

Claims 15, 16, 20-27, 31, 32, and 36-43 remain pending in the present application. Independent claims 15, 25, 31, and 41 have been amended to emphasize that an application is instantiated only after being authenticated by the security kernel during the preferred mode. Applicants respectfully submit that no new matter has been added to the application by the Amendment.

As a reminder, the present invention is directed toward the problem that for a computing device such as a portable computing device to be trusted in the context of a rights management architecture, the portable device and the processor thereon must be of a type that substantially completely prevents a content thief from performing nefarious acts that would allow obtaining of content therein in an unencrypted form or decryption keys. Thus, according to the present invention, the processor is a secure processor and is constructed to run only authorized code, and is operated to maintain a strict cryptographic separation between applications that may be instantiated thereon.

The secure processor is operable in a normal mode and a preferred mode, where a security kernel can access a locally accessible CPU key only during the preferred mode. The security kernel employs the accessed CPU key during the preferred mode to instantiate and/or authenticate a secure application such as a rights management system, a banking / financial system, etc. on the portable device. Importantly, whatever the application

may be, such application is not instantiated until authenticated by the security kernel during the preferred mode. The security kernel may automatically instantiate a particular secure application, may authenticate a secure application instantiated by another process, or may initially instantiate a secure chooser application that allows a user to select from one or more available secure applications on the portable device.

In any case, the accessed CPU key is typically a symmetric key that is employed by the security kernel to decrypt one or more encrypted security keys for the application instantiated. The CPU key is accessible only by the security kernel and only during the preferred mode, and is the key to unlocking or decrypting the secrets identified with each application, and therefore must be well-protected.

The Examiner has now rejected claims 15, 16, 20-27, 31, 32, and 36-43 under 35 USC § 103 as being obvious over Vu et al. (U.S. Patent No. 6,557,104) in view of Ginter et al. (U.S. Patent No. 5,892,900). Applicants respectfully traverse the § 103 rejection insofar as it may be applied to the claims as amended.

Independent claim 15 of the present application as amended recites a method for a secure processor to instantiate and authenticate a secure application thereon by way of a security kernel. In the method, the secure processor enters a preferred mode where a security key of the processor is accessible and instantiates and runs a security kernel. Thereafter, the security kernel accesses the security key and applies same to decrypt at least one encrypted key for the application, stores the decrypted key(s) in a location where the application will expect the key(s) to be found, and authenticates the application on the processor. Significantly, the application is instantiated only after the security kernel has authenticated such application..

Accordingly, the secure processor then enters a normal mode from the preferred mode after the security kernel authenticates the application and the application has been instantiated, where the security key is not accessible. Thus, the security kernel allows the processor to be trusted to keep hidden the key(s) of the application. The security kernel employs the accessed security key during the preferred mode to authenticate / verify the application prior to instantiation thereof.

Independent claim 31 recites the subject matter of claim 15, albeit in the form of a computer-readable medium.

As was previously pointed out, the Vu reference discloses a secure processor where, during run-time, an application requiring access to a secure service invokes a security routine that in turn invokes a security mode by way of a high-level security interrupt which cannot be otherwise invoked. As best set forth at column 5, lines 24-41, once in the security mode, a security function is invoked to access secrets and data from an otherwise inaccessible storage location, and the security function performs appropriate security functionality based on such secrets and data, such as for example encryption and decryption, password validation, user authentication, etc.

However, and significantly, in the Vu reference, the Vu application is already running and initiates the security function when services of a secure entity are required (see, for example, step 20 of Fig. 2). In contrast, the present invention as recited in the claims as amended does not even allow the application to be instantiated until the application has been authenticated by the security kernel during the preferred mode. Thus, the Vu reference does not disclose or even suggest an application that is instantiated only after being authenticated in the manner recited in claims 15 and 31.

The Examiner cites to the Ginter reference as teaching a cache or the like from which data is erased when transitioning between modes such that sensitive data employed during one mode is not available during a following mode. This notwithstanding, the Ginter reference like the Vu reference also fails to disclose or even suggest an application that is instantiated only after being authenticated in the manner recited in claims 15 and 31. Moreover, neither the Vu reference nor the Ginter reference teaches or even suggests transitioning between modes by way of a CPU reset, as is recited in claims 15 and 31. In Vu in particular, the transition is instead accomplished by way of an interrupt.

Independent claim 25 recites a method for a secure processor to instantiate one of a plurality of available secure applications thereon by way of a security kernel. In the method, a chooser value is set to a value corresponding to a chooser application upon power-up, and a preferred mode is entered upon a power-up CPU reset and instantiating the security kernel. The security kernel determines that the chooser value corresponds to the chooser application and therefore authenticates such chooser application, after which such chooser application is instantiated.

After the chooser application is instantiated, a normal mode is entered and the chooser application presents the plurality of available applications for selection by a user. Upon receiving a selection of one of the presented applications to be instantiated, the chooser value is set to a value corresponding to the selected application. Thereafter, a CPU reset is executed and the preferred mode is re-entered, and the security kernel is instantiated. The security kernel then determines that the chooser value corresponds to the selected application and therefore authenticates the selected application, after which such selected application is instantiated. Normal mode is then re-entered after the selected application is instantiated and

run. Thus, the security kernel allows the processor to be trusted to keep hidden a secret of the chooser application and a secret of the selected application.

Independent claim 41 recites the subject matter of claim 25, albeit in the form of a computer-readable medium.

Applicants reassert the arguments with respect to claims 15 and 31 to claims 25 and 41. In particular, Applicants again point out that the Vu reference does not disclose or even suggest a chooser application, a selected application, or any other application that is instantiated only after being authenticated in the manner recited in claims 25 and 41. Instead, and again, in the Vu system the application itself initiates authentication, and therefore cannot be instantiated only after such authentication. In addition, the Ginter reference like the Vu reference also fails to disclose or even suggest an application that is instantiated only after being authenticated in the manner recited in claims 25 and 41. Moreover, neither the Vu reference nor the Ginter reference teaches or even suggests transitioning between modes by way of a CPU reset, as is recited in claims 25 and 41.

In addition, Applicants again respectfully submit that the Vu reference is utterly silent about employing a chooser application, a chooser value, and the use thereof to securely choose and instantiate one of a plurality of applications on a secure processor in the manner recited in claims 25 and 41. In particular, the Vu reference does not at all disclose or even suggest switching between modes as recited to first load and then operate a chooser application, employ same to select a chooser value corresponding to a chosen application, and then load and operate a chosen application in the manner recited in claims 25 and 41.

Accordingly, Applicants respectfully submit that the Vu and Ginter references cannot be combined to make obvious the invention recited in claims 15, 25, 31, and 41 or any

**DOCKET NO.:** MSFT-0312/164268  
**Application No.:** 09/892,329  
**Office Action Dated:** January 18, 2006

**PATENT**

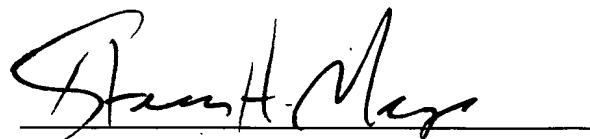
claims depending therefrom. Accordingly, and for all the aforementioned reasons,  
Applicants respectfully request reconsideration and withdrawal of the § 103 rejection.

**DOCKET NO.:** MSFT-0312/164268  
**Application No.:** 09/892,329  
**Office Action Dated:** January 18, 2006

**PATENT**

In view of the foregoing discussion, Applicants respectfully submit that the present application, including claims 15, 16, 20-27, 31, 32, and 36-43, is in condition for allowance, and such action is respectfully requested.

Respectfully submitted,

A handwritten signature in black ink, appearing to read "Steven H. Meyer", written over a horizontal line.

Steven H. Meyer  
Registration No. 37,189

Date: April 11, 2006, 2006

Woodcock Washburn LLP  
One Liberty Place - 46th Floor  
Philadelphia PA 19103  
Telephone: (215) 568-3100  
Facsimile: (215) 568-3439